

[About IEEE](#) | [IEEE Memberships](#) | [Products and Services](#) | [Conferences](#) | [IEEE Organizations](#) | [News](#) | [Home](#)**IEEE Xplore™**

Abstract

[Help](#) [FAQ](#) [Terms](#)

Welcome to IEEE Xplore™

- ☐ Home
- ☐ Log-out

Tables of Contents

- ☐ Journals & Magazines
- ☐ Conference Proceedings
- ☐ Standards

Search

- ☐ By Author
- ☐ Basic
- ☐ Advanced

Member Services

- ☐ Join IEEE
- ☐ Establish IEEE Web Account

Print Format

[SEARCH RESULTS](#) [PDF FULL-TEXT](#) [PREVIOUS](#) [NEXT](#)

## Eliminating the Z-Buffer bottleneck

- Knittel, G.; Schilling, A.

WSI/GRIS, Tübingen Univ., Germany

*This paper appears in: **European Design and Test Conference, 1995. ED8 Proceedings.***

On page(s): 12 - 16

6-9 March 1995

1995

ISBN: 0-8186-7039-8

IEEE Catalog Number: 95TH8058

Number of Pages: xxvii+611

References Cited: 2

INSPEC Accession Number: 5044342

---

### Abstract:

We present a solution to one of the fundamental problems in computer graphics hidden surface removal. In most 3D-graphics systems the hidden surface removal is done using the Z-**Buffer** algorithm. This method, however, requires one to perform a **read-modify-write** memory access for each visible pixel, which represents a performance limit. We introduce a novel SRAM cell, which incorporates the necessary logical units to perform the Z-**Buffer** algorithm on its own. Placed into the place of conventional DRAMs, almost any pixel rate can be achieved.

---

### Index Terms:

**buffer** storage; hidden feature removal; computer graphics; SRAM chips; memory architecture; Z-**buffer** bottleneck; computer graphics; hidden surface removal; SRAM cell; logical units; page register; pixel rate

[SEARCH RESULTS](#) [PDF FULL-TEXT](#) [PREVIOUS](#) [NEXT](#)

[Home](#) | [Log-out](#) | [Journals](#) | [Conference Proceedings](#) | [Standards](#)  
[Search by Author](#) | [Basic Search](#) | [Advanced Search](#) | [Join IEEE](#) | [Establish a Web Account](#)

Copyright © 2000 IEEE -- All rights reserved

# Eliminating the Z-Buffer Bottleneck

Günter Knittel, Andreas Schilling

WSI / GRIS

University of Tübingen, Germany

email: [knittelandreas]@gris.informatik.uni-tuebingen.de

## Abstract

*We present a solution to one of the fundamental problems in computer graphics, the hidden surface removal. In most 3D-graphics systems the hidden surface removal is done using the Z-Buffer algorithm. This method, however, requires to perform a read-modify-write memory access for each visible pixel, what represents a severe performance limit. We introduce a novel SRAM cell, which incorporates the needed logical units to perform the Z-Buffer algorithm on its own. Placed into the page register of conventional DRAMs, almost any pixel rate can be achieved.*

## 1 Introduction

A common representation of 3-dimensional objects in computer graphics applications is an approximation of their surface by a sufficiently large number of polygons, mostly triangles. Graphical output devices nowadays are almost exclusively *raster devices*, that is, the display area is comprised of a set of discrete pixels. Each pixel on the screen is associated with one entry in the display buffer (or *Frame Buffer*), which holds the color of the pixel. For display purposes, each triangle must therefore be decomposed into the set of screen pixels it covers, a process called *rasterization*. Current state-of-the-art rasterizer units are capable of producing in excess of 100M pixels/s, if fed with triangles at an appropriate rate.

The non-trivial task of determining the visible parts of an object can be solved by computing for each pixel a measure for the distance to the observer, called *Z-Value*, and by comparing that Z-Value to the Z-Value of the pixel previously generated for this screen address. This is the so-called *Z-Buffer Algorithm* [1],[2], which requires additional memory capacity to hold the Z-Values.

Thus, before a pixel at screen address (x,y) can be written, the value in the Z-Buffer at address (x,y) must be read and compared to the newly generated one. If the new pixel is nearer to the observer, its Z-value must be written into the Z-Buffer, and its color is stored in the Frame Buffer to be displayed on the screen. The new pixel is discarded if it was behind the old one.

Obviously, the algorithm bears an inherent memory bandwidth problem. A typical screen resolution of about 1M pixels and 16 to 32 bits per Z-Value prohibit the use of fast SRAM devices due to the high costs. Slow DRAM devices force the system designer to build highly-interleaved and expensive memory systems to keep up with the fast rasterizer units.

As a solution, we propose to integrate the compare logic into the Z-Buffer memory devices, and to perform the complete Z-Buffer algorithm on chip. In this way, the enormous internal bandwidth is available, and the read-modify-write-cycles are turned into merely write-cycles from an external point of view. In its basic configuration, the Z-Buffer only outputs the farther/nearer-flag as a control signal for the Frame Buffer.

However, the compare logic must be fast, compact and compatible to the DRAM technology to be easily integrated. In this paper we will describe a logic embedded SRAM cell, called *CBit cell*, capable of performing the Z-compare operation, and a memory architecture, called *ZRAM*, which incorporates this cell for extremely high pixel rates.

## 2 CBit cells

For explanation purposes, we assume a Z-resolution of 32 bits. Let's consider the logic embedded SRAM cell in Figure 1, which holds the MSB  $Z_{O31}$  of the old Z-value. It must be compared to the newly generated MSB  $Z_{N31}$ , which is put on the true and inverted bit-lines. The upper half of the schematic consists of a common 6-transistor CMOS static RAM cell. The remaining seven transistors perform the compare operation. They are mainly N-type, and therefore, the cell is called an N-type CBit cell (there is a corresponding P-type cell, which will be introduced later). The operation is as follows:

- Prior to any access, the write-signal WR and the select-signal  $S_{31}$  are held low, the nearer flag  $\overline{NN}$  is pre-charged high. Thus,  $\overline{S}_{30} = 1$ .
- An access starts by placing the incoming  $Z_{N31}$ -bit and its inverted value on the corresponding bit lines. Then,  $S_{31}$  is activated and  $\overline{NN}$  is left floating.

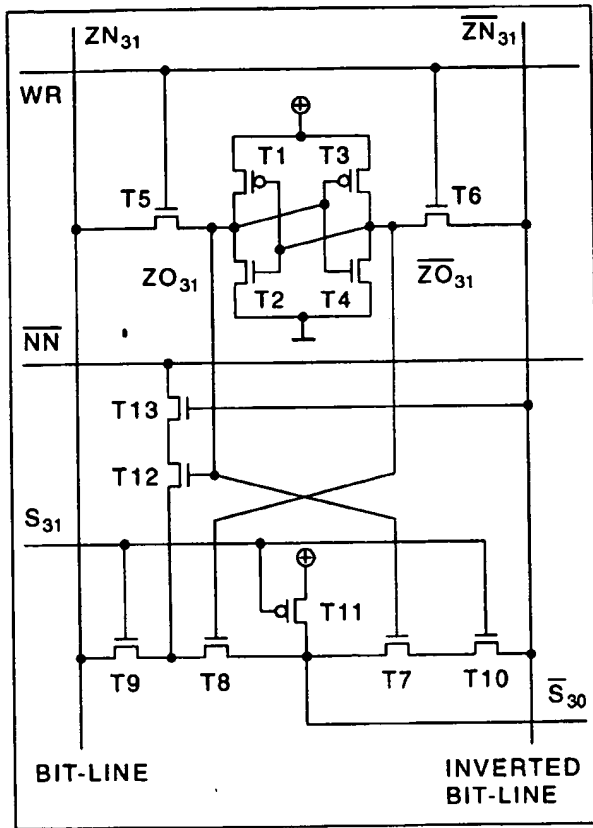


Figure 1: N-type CBit cell

This will produce logical values on the output lines as given in Table 1.

$ZN_{31}$	$ZO_{31}$	T7	T8	T12	T13	$\overline{NN}$	$\overline{S}_{30}$
0	0	OFF	ON	OFF	ON	1	0
0	1	ON	OFF	ON	ON	0	1
1	0	OFF	ON	OFF	OFF	1	1
1	1	ON	OFF	ON	OFF	1	0

Table 1: Functional behavior of N-type CBit cells

Thus, the  $\overline{NN}$ -signal goes active (low) if  $ZN_{31} < ZO_{31}$ , indicating at the same time that the new Z-Value  $ZN_{[31..0]}$  is smaller than the stored Z-Value  $ZO_{[31..0]}$ . In the case  $ZN_{31} > ZO_{31}$  the decision is made in the MSB as well. Only in the case  $ZN_{31} = ZO_{31}$  the next lower bit must be tested. Thus, the  $\overline{S}_{30}$ -line can be used to activate the CBit cell holding  $ZO_{30}$ .

However, we have to consider that the active level of the select signal has changed from the input to the output of the cell. This seems to be very impractical, and one could think about various arrangements of P-type and N-type transistors at the places T7 - T11. However, regard-

less what type the pass transistors are, they are good at passing a voltage level which does not switch on other pass transistors of the same type. An inverter would solve the problem, but it would also increase the propagation delay and the transistor count for each cell. Therefore, we construct a P-type CBit cell as shown in Figure 2, which is activated directly by the active low  $\overline{S}_{30}$ -signal.

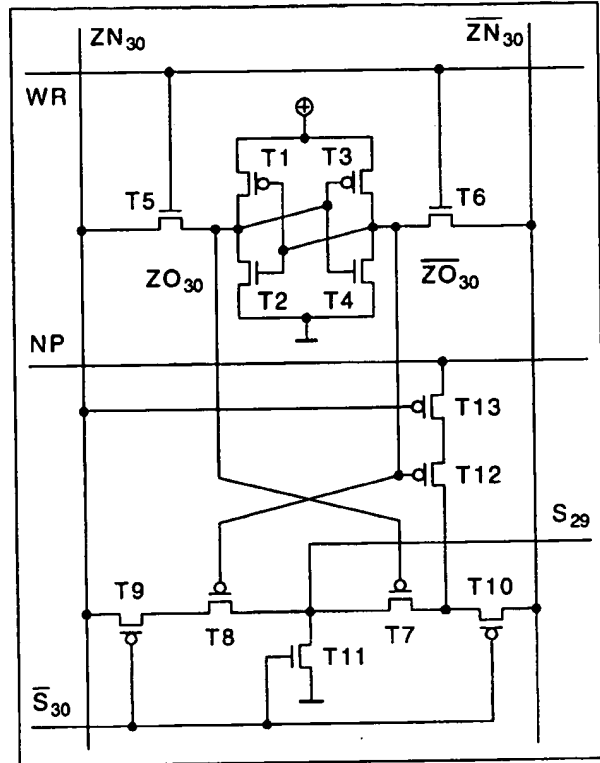


Figure 2: P-type CBit cell

Consequently, we have to introduce the line NP, indicating the nearer-condition for P-type CBit cells. It has an active high level and is pulled low prior to any access. NP and  $\overline{NN}$  are not connected to each other, and only one of them can be activated during a given access.

If selected, the internal state of the P-type CBit cell is shown in Table 2.

$ZN_{30}$	$ZO_{30}$	T7	T8	T12	T13	NP	$S_{29}$
0	0	ON	OFF	OFF	ON	0	1
0	1	OFF	ON	ON	ON	1	0
1	0	ON	OFF	OFF	OFF	0	0
1	1	OFF	ON	ON	OFF	0	1

Table 2: Functional behavior of P-type CBit cells

This time, the next lower CBit cell is activated by passing a high level on the  $S_{29}$ -line in the case  $ZN_{30} = ZO_{30}$ .

#### 4 Performance

A sample implementation of the circuitry was done at the IBM Development Laboratory at Böblingen using IBM's CMOS5L technology (3.3V, 0.5 $\mu$ m effective channel length). The following simulation results were obtained: the select-signal ripples through N-type and P-type cells in about 0.1ns and 0.15ns, respectively. After being selected, it takes 0.24ns for an N-type cell to activate the NN-line, and 0.56ns for a P-type cell to pull up the NP-line. Figure 5 shows the timing diagram of one N- and P-type CBit cell combination. The markers indicate delays as explained below.

Interval	Condition	Delay
V2 - V1, V4 - V3	$ZN_{31..30} = ZO_{31..30}$	0.25ns
V6 - V5	Nearer-Condition (N-type)	0.24ns
V8 - V7	Nearer-Condition (P-type)	0.66ns

In this implementation, a worst-case 32-bit compare operation takes about 4.4ns. However, the compare time can be brought well into the sub-nanosecond range without increasing the hardware expenses significantly. This is accomplished by breaking the select-chain into a number of shorter sub-chains, activating the select-signals of the CBit cells holding the MSB of each sub-chain simultaneously and combining the results. An example is shown in Figure 6, where the 32-bit chain is divided into 4 sub-chains, thereby reducing the overall delay time to approximately one fourth. The additional hardware expenses are as low as 12 transistors.

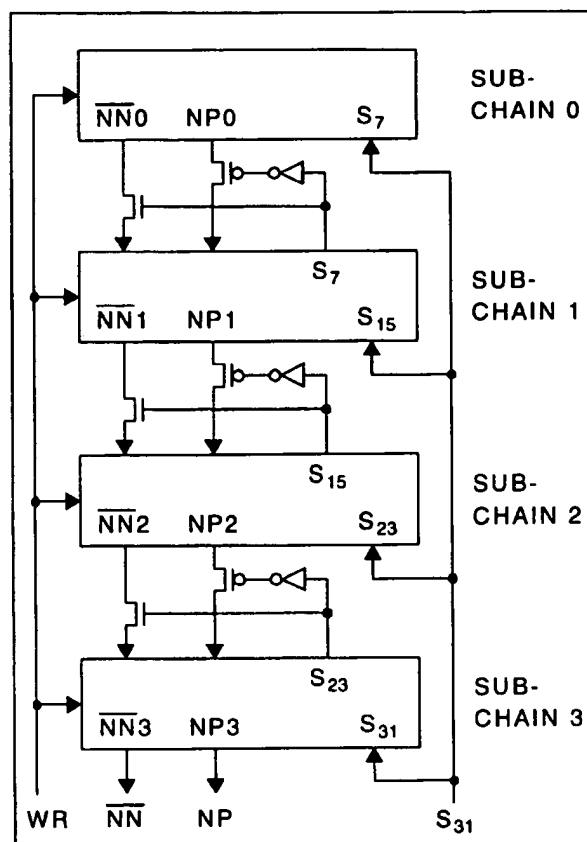


Figure 6: Speed-up by parallel operation of sub-chains

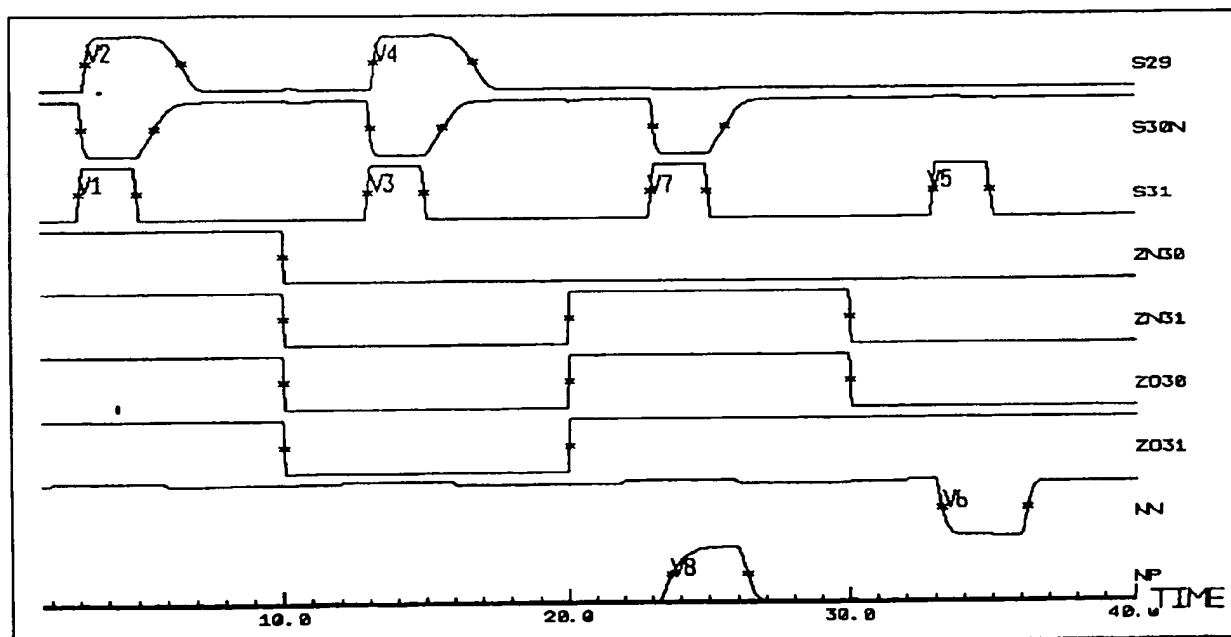


Figure 5: Timing diagram of one N- and P-type CBit cell combination

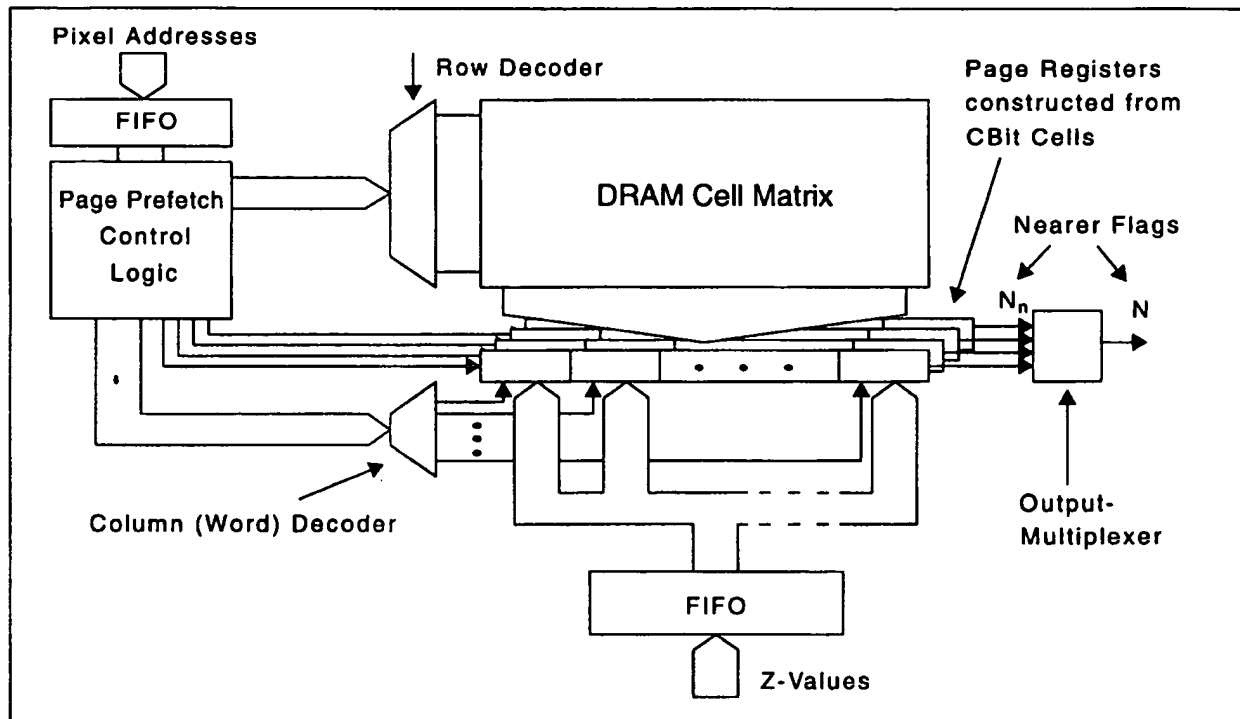


Figure 7: Architecture of the ZRAM-Chip

## 5 Architecture of the ZRAM-Chip

For the Z-Buffer, we propose to modify standard DRAM devices such that the page register is constructed from CBit cells. In the simplest configuration, the ZRAM chip has a 32-bit data interface for the Z-Values, and therefore accepts pixel addresses. The page register is organized such that 32 adjacent CBit cells hold the Z-Value of one pixel. Supposed the DRAM block is organized as 1Kx1K memory cells, the appropriate page register stores the Z-Values of 32 pixels. Most rasterizers operate screen line oriented, so that assigning screen line fragments to DRAM pages will give a high percentage of page hits. Nevertheless, any page fault will result in a severe performance loss due to the short processing time of one Z-compare operation. Thus we propose to install multiple page registers, and to use a page prefetch mechanism. The operation is as follows: upon receipt, Z-Value and pixel address are stored in FIFO memories placed onto the ZRAM as well. While performing the appropriate page access, new data is allowed to enter the chip, and succeeding page faults are detected. As soon as operation starts with the just loaded page register, a new page access is initiated, and so forth. A block diagram of the proposed architecture is shown in Figure 7. For simplicity, we assume the DRAM device being built as one single memory block. Due to the pipelined operation, the nearer-flags are handed out to the Frame Buffer Controller with a certain latency.

## 6 Conclusion

We presented a memory design which represents the ultimate solution to the notorious Z-Buffer Bottleneck. No new technology is required, and additional hardware expenses are small. Not described in this paper (since beyond the technical focus of the Conference) are further possibilities of this design, such as supporting Anti-Aliasing on sub-pixel resolution and further increasing the pixel rate by placing parts of the rasterizer unit onto the ZRAM as well.

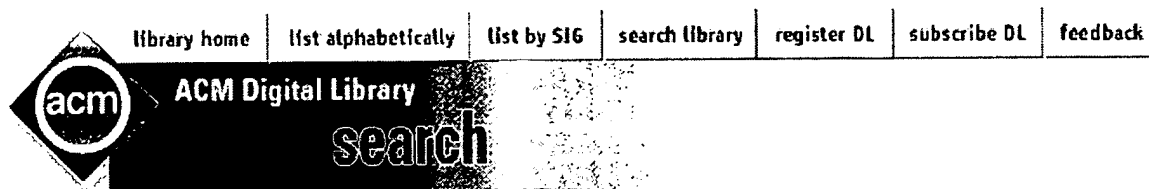
## 7 Acknowledgments

This work was supervised by Prof. Straßer and was done for the MONOGRAPH project, supported by the CEC's ESPRIT programme.

We owe very much to W. Löhlein from IBM Development Laboratory at Böblingen, who gave us valuable insight into memory technology. He implemented the circuitry and provided the timing data.

## 8 References

1. E. Catnull, "A Subdivision Algorithm for Computer Display of Curved Surfaces", Ph.D.-Thesis, Computer Science Department, University of Utah, December 1974
2. W. Straßer, "Schnelle Kurven- und Flächendarstellung auf graphischen Sichtgeräten", Ph.D.-Thesis D83, Technical University of Berlin, Germany, 1974



# Search the Digital Library

## Search Articles:

Terms:

read modify write buffer graphics

☒ all words ☐ any words ☐ exact phrase ☐ subject  
☐ expression (☐ stem)

In Fields:

<input checked="" type="checkbox"/> Title (71,013)	<input checked="" type="checkbox"/> Reviews (2,654)
<input checked="" type="checkbox"/> Full-Text (62,871)	<input checked="" type="checkbox"/> Index Terms (54,809)
<input checked="" type="checkbox"/> Abstract (21,006)	(Number of articles)

Authors:

☒ all names ☐ any name ☐ expression (☐ soundex)

## Limit Your Search To:

Publication:

All Journals and Proceedings

Published  
Since:

Month

Year

Published  
Before:

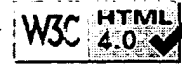
Month

Year

search

[ [Help](#) ]

The Digital Library is published by the Association for  
Computing Machinery. Copyright © 2000 ACM, Inc.

[library home](#)[list alphabetically](#)[list by SIG](#)[search library](#)[register DL](#)[subscribe DL](#)[feedback](#)

[About IEEE](#) | [IEEE Memberships](#) | [Products and Services](#) | [Conferences](#) | [IEEE Organizations](#) | [News](#) | [Home](#)**IEEE Xplore™**

Search

[Help](#) [FAQ](#) [Terms](#)**Welcome to IEEE Xplore™**

- ☐ Home
- ☐ Log-out

**Tables of Contents**

- ☐ Journals & Magazines
- ☐ Conference Proceedings
- ☐ Standards

**Search**

- ☐ By Author
- ☐ Basic
- ☐ Advanced

**Member Services**

- ☐ Join IEEE
- ☐ Establish IEEE Web Account

Your search matched **3** of **697946** documents.Results are shown **15** to a page, sorted by **publication year** in **descending** order.You may refine your search by editing the current search expression or entering a new one the text box. Then click **Search Again**.

read &lt;sentence&gt; modify &lt;sentence&gt; write &lt;sentence&gt; buffer

**Results:**Journal or Magazine = **JNL** Conference = **CNF** Standard = **STD****1 Eliminating the Z-Buffer bottleneck***Knittel, G.; Schilling, A.*European Design and Test Conference, 1995. ED&TC; 1995, Proceedings. , 1995  
Page(s): 12 -16[\[Abstract\]](#) [\[PDF Full-Text\]](#) **CNF****2 Accurate rendering by subpixel addressing***Lathrop, O.; Kirk, D.; Voorhies, D.*IEEE Computer Graphics and Applications , Volume: 10 Issue: 5 , Sept. 1990  
Page(s): 45 -53[\[Abstract\]](#) [\[PDF Full-Text\]](#) **JNL****3 An on-chip 72 K pseudo two-port cache memory subsystem***Chuang, S.C.-M.; Mukherjee, T.; Bracer, G.; Litten, S.; Peters, M.; LeBlanc, T.; Taroni, G.; Akrou, C.*VLSI Circuits, 1990. Digest of Technical Papers., 1990 Symposium on , 1990  
Page(s): 113 -114[\[Abstract\]](#) [\[PDF Full-Text\]](#) **CNF**[Home](#) | [Log-out](#) | [Journals](#) | [Conference Proceedings](#) | [Standards](#)  
[Search by Author](#) | [Basic Search](#) | [Advanced Search](#) | [Join IEEE](#) | [Establish a Web Account](#)

Copyright © 2000 IEEE -- All rights reserved



[About IEEE](#) | [IEEE Memberships](#) | [Products and Services](#) | [Conferences](#) | [IEEE Organizations](#) | [News](#) | [Home](#)



# IEEE Xplore™

Search

[Help](#) [FAQ](#) [Terms](#)

## Welcome to IEEE Xplore™

- ☐ Home
- ☐ Log-out

## Tables of Contents

- ☐ Journals & Magazines
- ☐ Conference Proceedings
- ☐ Standards

## Search

- ☐ By Author
- ☐ Basic
- ☐ Advanced

## Member Services

- ☐ Join IEEE
- ☐ Establish IEEE Web Account

Your search matched **5** of **697946** documents.

Results are shown **25** to a page, sorted by **publication year** in **descending** order.

You may refine your search by editing the current search expression or entering a new one the te  
Then click **Search Again**.

read <sentence> modify <sentence> write <paragraph> suspend

**Search Again**

## Results:

Journal or Magazine = **JNL** Conference = **CNF** Standard = **STD**

### 1 **A 10 Mb frame buffer memory with Z-compare and A-blend units**

*Inoue, K.; Nakamura, H.; Kawai, H.*

Solid-State Circuits, IEEE Journal of , Volume: 30 Issue: 12 , Dec. 1995

Page(s): 1563 -1568

[\[Abstract\]](#) [\[PDF Full-Text\]](#) **JNL**

### 2 **Eliminating the Z-Buffer bottleneck**

*Knittel, G.; Schilling, A.*

European Design and Test Conference, 1995. ED&TC; 1995, Proceedings. , :

Page(s): 12 -16

[\[Abstract\]](#) [\[PDF Full-Text\]](#) **CNF**

### 3 **A 9-ns, 1.4-gigabyte/s, 17-ported CMOS register file**

*Jolly, R.D.*

Solid-State Circuits, IEEE Journal of , Volume: 26 Issue: 10 , Oct. 1991

Page(s): 1407 -1412

[\[Abstract\]](#) [\[PDF Full-Text\]](#) **JNL**

### 4 **Accurate rendering by subpixel addressing**

*Lathrop, O.; Kirk, D.; Voorhies, D.*

IEEE Computer Graphics and Applications , Volume: 10 Issue: 5 , Sept. 1990

Page(s): 45 -53

[\[Abstract\]](#) [\[PDF Full-Text\]](#) **JNL**

### 5 **An on-chip 72 K pseudo two-port cache memory subsystem**

*Chuang, S.C.-M.; Mukherjee, T.; Bracer, G.; Litten, S.; Peters, M.; LeBlanc, Taroni, G.; Akrou, C.*

VLSI Circuits, 1990. Digest of Technical Papers., 1990 Symposium on , 1990

Page(s): 113 -114

[\[Abstract\]](#) [\[PDF Full-Text\]](#) **CNF**



Creation date: 03-16-2005  
Indexing Officer: LCHEO - LEE CHEO  
Team: OIPEBackFileIndexing  
Dossier: 09280250

Legal Date: 07-05-2001

No.	Doccode	Number of pages
1	SRNT	4

Total number of pages: 4

Remarks:

Order of re-scan issued on .....